Chapter 9

''Walk-in'' Interfaces and Internet Applications

Remember the airline reading light we discussed earlier? Well, as with everything, the machines that passengers interact with on today's airplanes have grown ever more complex. In this chapter, we look at the new personal entertainment system available on many aircraft. We start by pressing the video button on the personal control unit in the armrest (figure 9.1). The blank screen awakes with a flicker, but not much else. You hit the video button again but still nothing. That's not very entertaining. You play with the brightness controls thinking maybe that is the problem—but it's not. So you try every button, hoping that something might turn up on the screen. Not today.

Maybe there just aren't any movies on this flight—but two rows down someone else's screen is vivid with action. You might consider calling the flight attendant but you decide to try it again on your own. Maybe the in-flight magazine will explain how this entertainment system works, but if it does you cannot find where. You conclude that either the screen is broken or maybe you are just inept when it comes to technology.

Actually, nothing's broken, and you're not inept. The screen is just fine and all nine movies are playing. It's only that this user-interface has a problem, or to be more precise, the people who designed it were just not sensitive enough to the subtleties of how people interact with kiosks, or "walk-in" types of devices—you know, the Automatic Teller Machines (ATM) outside banks, electronic check-in devices at the airport, and credit-card payment devices at gas stations. We are expected to use such "walk-in" machines without any prior knowledge; there is no user manual, and forget about any training.

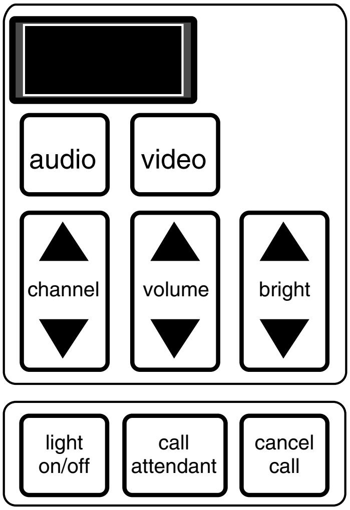

Figure 9.1. The armrest control unit.

Inside the Machine

So let's examine this in-flight entertainment system to understand the kind of problems that exist with these "walk-in" interfaces. Figure 9.2 shows you the states and modes of the system. When you are seated in your cushiony seat at the start of the flight, the system is IDLE and the screen, naturally, is BLANK. You operate the entertainment system by bringing up VIDEO, or AUDIO, or a FLIGHT MAP showing the aircraft position and some geography.

When you select VIDEO, you not only engage this mode but also trigger three of its associated reference values: the video (movie) channels, the brightness, and volume. All three reference values must be in coordination for VIDEO to become active. Here, we shall focus only on the channel reference value, which, as you can see in figure 9.3, runs concurrently to VIDEO. When you select VIDEO, the system wakes up as follows: the on screen turns silver and at the same time, the initial video channel is set to 0 (note the little arrow pointing at 0).

And this is where the problem arises. The A mode, as we know from chapter 7, carries with it a variety of reference values. For us to see the movie, we need to select VIDEO and also a video channel (because the initial channel, "0," has nothing in it). Unless you manually select an active channel (1-9), you will not see anything. You can hit the "video" button as many times as you wish, but it won't get you anywhere because you will just loop around to see more and more empty silver screens.

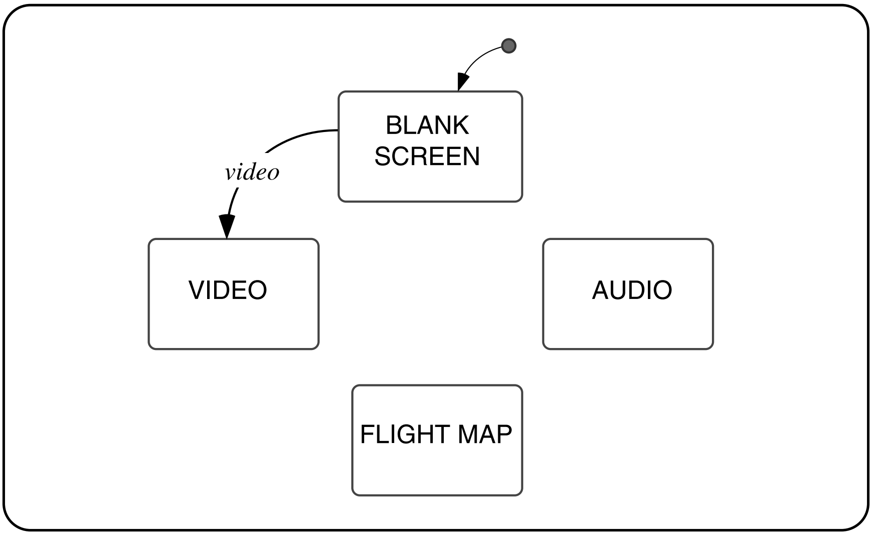

Figure 9.2. Four basic modes of the flight entertainment system.

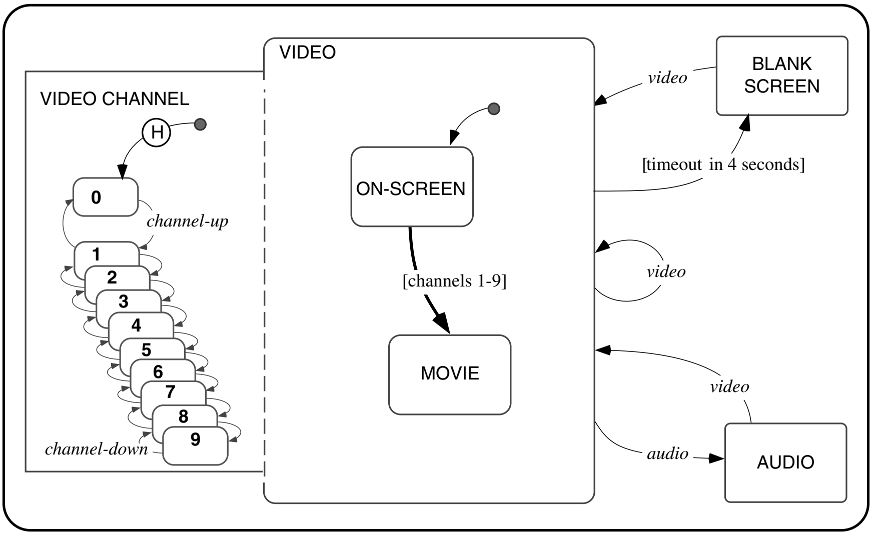

Figure 9.3. Inside the VIDEO mode.

If you are patient and hold off from hitting the "video" button endlessly, waiting for the movie to come on—this system will confuse you more, because in four seconds the screen will automatically turn blank. As shown in figure 9.3, there is a four-second timeout transition out of ON-SCREEN back to BLANK SCREEN state. The only way to see a movie is to press the channel (up or down) button, after you select VIDEO. But nobody tells you that. And it doesn't help that when VIDEO is selected, the little display on the armrest control panel turns dark. If it would display 0, then perhaps we would get a clue.

So why must you press the channel button for the device to transition to a movie? Is it reasonable to expect that people will immediately press "channel" after "video"? Maybe; the evidence, however, shows that many don't.

Initialization

The problem with the in-flight video system starts with the way this system initializes—namely, the initial mode and initial reference values. And then, of course, there are the user's expectations (such as seeing some kind of a screen image after selecting "video") that are violated. Top all this with poor feedback, and you get the kind of confusion that makes us, at times, hate technology.

The underlying concepts here—which are rather subtle, yet present in every device we use—are important to understand: we already know that every machine must start up in some initial configuration. These initial settings are pre-set and programmed by the people who designed the device. The initial configuration for this machine (after turning it on by hitting "video") is ON-SCREEN and CHANNEL 0—a mode and reference value combination—that appears on the screen as silver snow. The passenger who uses the device, however, expects to see a video image. When he doesn't see a movie after pressing "video," he gets a bit confused. After a couple of unsuccessful tries, frustration enters.

A sure way to deal with initial mode and reference value settings is to display and indicate them to the user. But, unfortunately, in this device the initial channel setting ("0") is not shown because the channel indicator on the armrest turns blank. One way to fix this design problem is to make channel 1 (or any other active channel) the initial setting. If that's not available, then perhaps a video image, which will indicate to the user that the screen is indeed functioning. A more comprehensive approach is to consider the initial status (such as the entry screen) of any device, and especially "walk-in" devices, as an "anchor." The initial screen must provide accurate feedback to the user about the working of the device. It can also be designed in a way that it provides the user with information/instructions on how to proceed (and perhaps even some troubleshooting tips).

Mode and Reference-Value Settings

Any time we change a mode of a device, the new mode carries with it one or more reference values. Sometimes these reference values are the same as the previous mode (e.g., the volume stays the same when we switch from VIDEO mode to AUDIO and back). Sometimes they change as a consequence of the mode switching. For example, when we change the mode of a microwave from COOK to POPCORN, the power setting (a reference value) changes to high, no matter what it was before. In COOK mode, the user must enter the cooking time, which is a changeable reference value; in POPCORN mode, the time is pre-set to 4 minutes and 20 seconds.

The issue here is that each mode carries with it a variety of reference values. Understanding this mode and reference-value marriage, and how this relationship changes over time, is an important aspect of user interaction with automated devices.

History Settings

Sometimes, the reference value(s) of a mode depends on the past setting. Consider the following scenario from the in-flight entertainment system: we select a video channel, say 3, and then decide to listen to some angelical Bach on audio channel 8. Afterwards, we decide to watch some earthly video. When we return to VIDEO mode, the channel setting remains at 3. That's what we all expect, right? This way of automatically re-selecting a reference value (channel 3), once we are back in VIDEO mode is called *history* and is depicted in figure 9.3 with an (H). The little arrow shows that on initialization, the reference value is channel 0—however, on any subsequent entries to VIDEO mode, the last setting (e.g., channel 3) is the one that is re-selected.

This history setting can be quite helpful to the user. It saves time and also reduces the amount of interaction with the device. If you look around you will see many examples of this. Your car's audio system, for example, has it. Here is a scenario that I'm sure is familiar: you play your favorite CD, and then decide to tune in to the news. After the newscast is over, you go back to your music. Your CD continues playing from exactly were you left it. But try that on your VCR/TV system at home and you will see that the video continued playing and when you return the movie is way beyond where you left it. On DVD players, however, when you switch modes (e.g., from DVD to TV) the DVD player stops automatically, and then resumes automatically from the spot you left off.

Both good and bad uses of the history feature are seen on the Internet. You go online to get some service and you are asked to fill in your personal information. Then for some reason you click out of the page, and then return back. Will the page contain all your previously entered data, or has it been erased, forcing you to start over again?

Default Settings

Another design problem with the in-flight entertainment system is the default transition out of ON-SCREEN. Here, when the user fails to select a channel (1–9), the screen automatically turns blank. This unfortunately gives the impression that the device has failed, when in fact the system is working just fine. To many of us, this is the last straw—from here we go immediately into the usual litany of self-criticism: "I'll never understand, I'm not good with these things, I hate machines," and on and on.

Like initializations, defaults are also pre-set actions that are programmed by the designers of the system. The term *default* comes from its legal meaning of failing some timely obligation—a loan, an appearance in court, a duty. Defaults exist in every interaction—human-human, human-machine, as well as when machines interact among themselves (such as during a secure interaction between computers). We may tell a friend "Meet me by the subway's entrance at 4 P.M., but if I'm not there in 20 minutes, go home and I'll call you later." *Go home* is the default action and 20 *minutes* is the time to wait it out.

We find defaults when we interact with businesses. Have you noticed those ads about a "month-long free trial" of a weekly magazine? You give them your credit card number and you start receiving the magazine. You can cancel the subscription anytime. However, if you don't cancel by the end of the trial period, *by default* your credit card gets (automatically) charged. Here the default action is to extend the subscription service. An example in reverse, with respect to the default action, is frequent-flier programs: when you reserve a seat with your hard-earned miles, the reservation will cancel, *by default*, if you don't book it within two weeks. Here the default action is to cancel the service (because it is a perk).

In human-machine interaction, default conditions are in place when the user fails to interact—the machine waits for some user interaction, and when it does not come, the system takes control. The classic example is when your ATM machine "eats" your card if you fail to remove it within a minute or so after the transaction is over. Defaults are also in place when the user places contradictory demands on the machine. By default, the machine reacts to one demand and ignores the other. We all understand defaults and are quite accustomed to them, yet they do pose a challenge for designers. Why? Because unlike initial settings, which you can always display and indicate what is the current mode and setting, default transitions will happen in time. It is not trivial to consistently inform the user what will be the consequences of his or her *inaction*. For machines that people own or use for work you can perhaps train and teach the user about defaults and/or describe them in the user manual. But what do you do when it comes to "walk-in" devices where training and manuals are not an option?

When we are dealing with walk-in devices, where the range of user skills in operating the device is huge, the designer has to be very careful. Any expectations on part of the designers about how the casual user "will" interact with a system are merely conjunctures. If there is an available path that the user can take—no matter how unlikely—there will be someone (or more than one) that *will* take it!

This is where models of user interactions, such as the ones described in this book, are useful. Once you have a model that details user interaction, undesirable paths can be eliminated or blocked off, while desired paths may be broadened and illuminated. It is further possible to eliminate much confusion by providing the users with "anchors," like screens that are inserted to help the user navigate his or her way along the system. A well-designed initialization screen can be used as an anchor. Defaulting back to such "anchors," whether it is the initial screen or some other screens that provide feedback on how to proceed, can also help.

To conclude this section, please note that the four concepts discussed here—*initial mode setting, initial reference-values, history settings*, and *default settings*—are all too often confused and used interchangeably. By understanding them and the subtle differences between them, it is possible to provide users with better feedback, and avoid the kind of design problems encountered in the in-flight entertainment system.

Visitors on the Internet

We are now ready to move on along our tour of "walk-in" interfaces. With only a few exceptions, all Internet interfaces are designed for immediate, no training necessary, use. Yet it is crucial for e-businesses that customers stay and not flee with frustration. It is also important that all of us—young, old, computer literate or novice—will be able to interact with Internet applications, otherwise we will be locked out of important information and resources. But how many times have you found yourself lost inside a web site? How many times have you had to retype your personal information, address and credit card number over and over, and wondering if the transaction did in fact take place or not—or if you will be charged twice? Why are so many Internet sites difficult to use?

The World Wide Web (WWW), or more accurately the idea behind what we now know as the WWW, first surfaced at CERN, the world-renowned physics research facility on the Swiss-French border, in 1989. For years, scientists and engineers have used information highways to transfer data files. The foundation was in place, but it lacked an interface that would make the interaction straightforward and usable. It was only with the first browser, Mosaic, developed by Marc Andreessen at the University of Illinois, that suddenly there was a real interface for looking at the underlying information. Mosaic, and then the Netscape browser, provided a graphic interface to the Internet in the same way that the Macintosh desktop provides a window into the working of the desktop computer.

We begin by considering an Internet application that is part of a small information system. The application allows employees of a large research and development institution to log, in advance, the visitors that they are expecting. Many people get confused while using this Internet application. On the machine's side, user confusion and resulting errors cause disorders in the databases, demanding frequent and costly maintenance to the software system. So let's go in and figure out for ourselves what the confusion is all about.

The initial page, VISITOR REQUEST MENU, has a small menu that allows you to "submit" a new visitor request and "browse" current requests, and "browse or copy" expired requests (see the top screen in figure 9.4). It looks simple. We start by clicking on the "submit" button. We land on a page labeled "visit information." Here you patiently enter dates of arrival, dates of departure, time of day, the purpose of the visit. In addition, they ask for your telephone number, your office number, an alternate person to call if you are not there, and so on. Now you're ready to "continue to the next step."

If, for whatever reason, you have tried to cut corners and skip entering the required information, the Web-warden will catch you. Conditional (C) on skipping required information, it displays an error message, a big white banner titled "Sorry, Error #185, you did not enter all the required information," telling you that you've skipped over this or that item. Otherwise, you are welcomed to the next page, which is titled "visitor submission" in figure 9.4.

On the VISITOR SUBMISSION page (at the bottom of the figure), we enter the visitor's information: first name, middle name, last name, affiliation, and so on. At the bottom of the screen there are three buttons: "save visitor," "show visitors," and "done." The "done" button seems most appropriate because we think we are. And as you can see on the right side of figure 9.4, a large page comes up. "Do you want to save visitor information before exiting"? But of course. Press the "yes" button and you are back to the visitor-request main menu, which is where we started from. And since there are eight dignitaries in this group of visitors, which means quite a lot of data entry, we need to repeat this cycle over and over: The visit information, the visitor information, and so on. Done. Back to the main menu. And then the same visit and visitor information again. Done. On and on for each visitor.

But wait. Actually, there is no need for such arduous work. As it turns out, there is a much easier and quicker way of entering visitor's information. The alternate route is shown in figure 9.5. If you click the "save visitor" button,

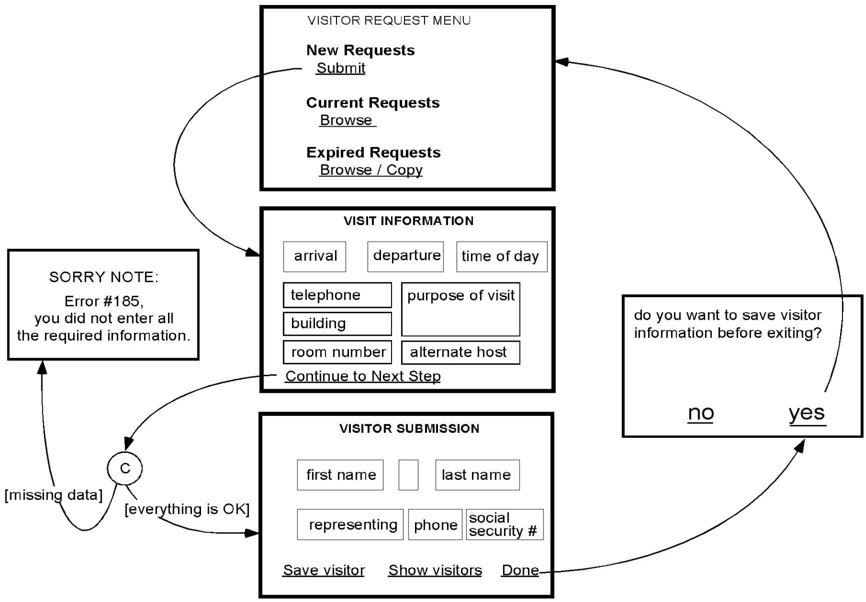

Figure 9.4. Visitor Request main menu and subsequent add/submit pages.

instead of "done," you immediately get another "visitor submission page." Fill it up with the visitor's name, click "save visitor" again, and get another VISITOR SUBMISSION. And so on and so on. This way you can do the data entry in a flash; there is no need to go back all the way to the main menu and there is no need to keep entering the repetitive information about the dates, the time, the purpose, etc., of the visit, which, of course, are common to all the visitors in the group.

Analysis

How would you even know that this labor-saving path is available? There are no cues for it on the interface. The underlying structure of this visitor request process, which can make life in the data entry lane a bit easier, is hidden. In figure 9.5, you will note that the "visit information" page is like the parent and the visitor information pages are the siblings. Once you fill in the visit information, each visitor inherits this information from the parent, and there is no need to repeat it. Part of the misunderstanding lies in the fact that there is a subtle difference between *visit* and *visitor* here: the word *visit* in this Internet application relates to the global information of the visit (the dates, purpose, host, etc.); *visitor* relates to the individual (first name, middle, last, and so on). If you click for the "Help" page and read it carefully, you might find some explanation. But let's be honest, how many times have you gone into and

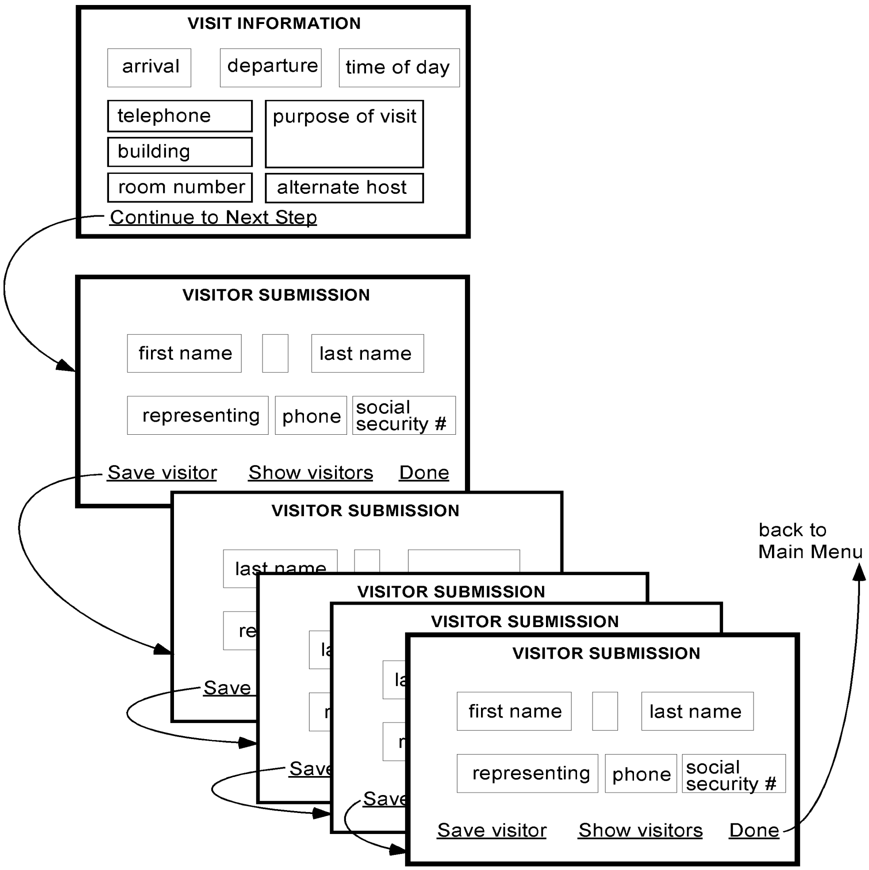

Figure 9.5. The structure of the additional visitor submissions.

carefully studied the help information? And even if you read it, you quickly find that it is not clear and straightforward given the subtle difference between visit and visitors.

Whether you do take the time to read the "Help" page or not, the heart of the problem lies with the hidden structure of this site. The hierarchical relationship between visit (parent) and visitors (siblings) is not portrayed on the page, nor is there any cue or hint of this underlying structure. The visitor information site takes you by the hand and moves you from one page to another. If you follow—and what else can you do?—you'll end up doing much more work than necessary. It is very similar to getting travel directions: "Drive three blocks, then make a right, then turn left at the fifth traffic light, and continued until you reach a dead end." The problem with such navigation, as we all know, is that when the instructions are incomplete—if you miss a turn or make a mistake in counting traffic lights—you are doomed. The only road to salvation is to find the starting point (if you are lucky) and begin again.

Navigation

What is common to all good navigators of land, sea, and air is their ability to understand and take advantage of the structure of their environment. Walking down Seventh Avenue in New York City, an urban "navigator" knows that Sixth Avenue is running parallel on the left and street numbers are descending. Similarly, a desert caravan leader knows that sand dunes run parallel and that a change in their shape may indicate the presence of an oasis. Ocean navigators in the age of discovery closely observed the color of the ocean, the smell of the air, and the shape of distant clouds as indicators of nearby land. The key to successful navigation is the ability to "read" the terrain. Once you know how to read this structure, you can fix the current location and simply proceed. Without an ability to read the structure of the environment, any navigator is lost.

We tend to think that navigation, of any kind, is all about getting there. But really it is about the process of finding one's own way within a certain context and making constant adjustments. The trick to any navigation is to have a reasonably good description and understanding of the underlying structure. Once you have that, you can deal with any contingencies. So if you're driving down Seventh Avenue and your destination is Seventh between 53rd and 54th Streets, and there's some kind of a jam on Seventh, you can slip onto Sixth, drive down a few blocks, and then return back to Seventh Avenue. We all understand that intuitively. But if you put a nomad, who may be a wonderful navigator in the desert, in New York City, there's no chance that he will be able to perform the seemingly (for us) simple maneuver.

Solutions and Exit

So what can be done to improve this visitor information site? One immediate solution is to provide, in the "Help" page, the structure of the application (depicted in figure 9.5). Better yet is to design the pages so that it will be clear to the user that there are two options: if this is a single-person visit, then just complete the "visit information" and return to the main menu; if there is more than one visitor, then there is another route to take. The good news is that when the user is made aware of the structure, things become much easier. It is possible to recover from mistakes, perhaps even finding a novel path when the unexpected occurs.

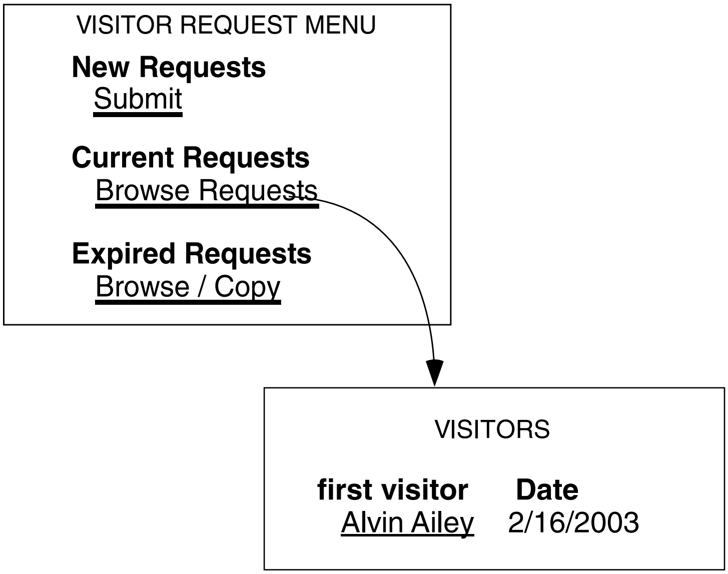

Figure 9.6. Where have all the flowers gone?

OK, so now we are finally done. Now we want to exit. But how? If we click on the "save" button, we're going to get another visitor page, which is definitely not what we want. So we hit the "done" button and get the page with the question "Do you want to save visitor information before exiting?" Yes—but what about the rest of the visitors? Will they be saved too? Well, there are no other choices here. Click "yes," and we are back in the main menu. Click on the "current requests" to make sure that all visitors are saved in there (figure 9.6), and all you see is . . . one visitor on the list!

Where are all the other visitors that you painstakingly entered? Only the first one is listed. What on earth happened to the rest?

Actually, all of them are in there and well saved—it's just that the page doesn't show it. If you click on the first visitor, you can see the "visit information." Press on the "show visitors" button on the lower left side of the screen, and the rest of the visitors show up on the screen (figure 9.7).

Here again, there's actually an underlying structure: the first visitor is like an anchor for the rest to follow. You can see this relationship in figure 9.7. The VISITORS page only contains the name of the first visitor—there is absolutely no clue to this abridged scheme on the page.

The effect of this problematic interface, which obscures the underlying structure, on user interaction is threefold. One, the user is uncomfortable with the site: you get unclear messages after clicking buttons that seem reasonable to click (e.g., the "done" button leads to more questions), and you find yourself

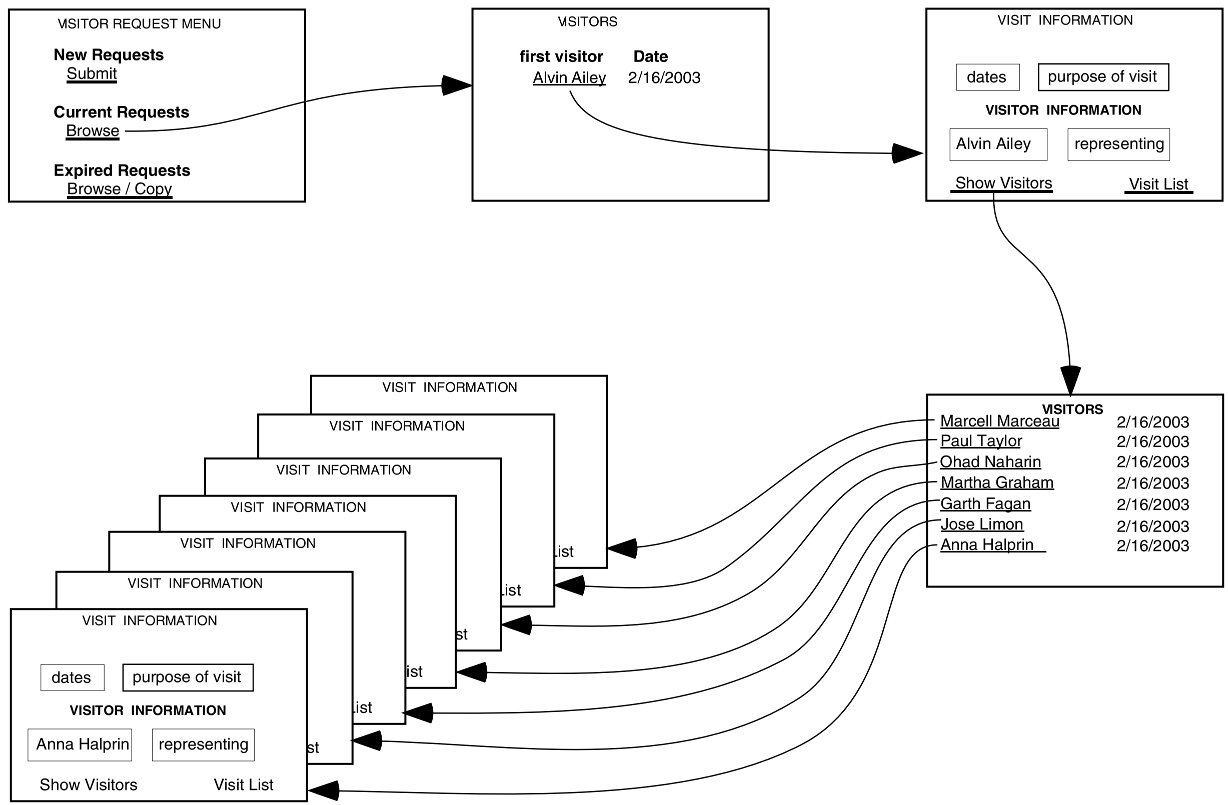

Figure 9.7. Checking the status of the visit.

baffled by what you get (how come only the first visitor has been saved?). Two, most people end up spending far more time than needed in doing this registration task. Worst of all, since nobody really cares to understand the workings of this application—most people just want to get it done, and leave—the problem is never aired. But then, of course, you end up coming back, to the same confusion and inefficiency the next time you have to use the application.

To conclude, the structure of the web site is the foundation of the interaction. The relevant structure must be made apparent to the user, especially when it comes to "walk-in" interfaces. In the beginning, ATM machines were our first encounter with such interfaces. But now we find walk-in interfaces when we buy bus tickets, pay for parking, and call up customer service ("listen carefully as our menu selections have recently changed"). But most of all, almost any web transaction is a walk-in interface. In the not-so-distant-future, we will see more of them in supermarkets, at the dry cleaners, at the post office, and believe it or not, also at the pizza parlor. We will miss the familiar face behind the counter and will have to get used to the interface. But so that all of us—the young, the old, and even the ones that define themselves as "computer illiterate"—will be able to shop, interact, and get errands and chores done, we will need much more sophisticated interfaces. And sophistication is not about fancy colors, flashing icons, and an endless amount of buttons to zing from one place to another—it's about understanding the user's tasks and expectations and building the interface to fit.

Vicente in his book *The Human Factor: Revolutionizing the Way People Live with Technology* (Knopf Canada, 2003; see the auto-clock story in chapter 7).

Chapter 8

Sailors are a superstitious lot, and so are executives of cruise lines. The reputation of a grounded ship does not sail very far. And hence the good ship *Royal Majesty*, once the jewel of the fleet, was eventually sold to Norwegian Cruise Lines, where she began sailing under the name *Norwegian Majesty*. In 1999 she voyaged to Bremerhaven, Germany, where she was hauled up and sliced in the middle. A 112-foot midsection was inserted, giving her an overall length of 680 feet and erasing all physical signs of her embarrassing grounding.

In August of 2001, the Asia Rip (AR) buoy (which was confused by the crew of the *Royal Majesty* with the entrance buoy to the Boston traffic lanes), was "decommissioned" and removed from location. The long and heavy chain that connected her to the wreck below was cut, and the buoy was towed back to shore. Notices to Mariners, issued by the U.S. Coast Guard and dated August 28, 2001, instructs mariners and navigators to cross out Asia Rip from all charts and piloting books.

The factual account for this chapter came primarily from a 1997 National Transportation Safety Board (NTSB) report about the accident and a variety of maritime publications. (The full report, titled *The Grounding of the Panamanian Passenger Ship Royal Majesty on Rose and Crown Shoal near Nantucket, Massachusetts on June 10, 1995,* can be obtained from the NTSB or downloaded from their web site.) The sequence of events and interactions between the bridge officers and the captain is based on the crews' testimonies following the accident. The description of the routine sea duties and some of the non-technical aspects of this story come from my own experiences as a bridge officer on naval and commercial vessels.

The Global Positioning System is a constellation of 24 satellites orbiting the Earth. The GPS unit obtains its distance from at least three satellites, and then triangulates its exact position on the Earth. Nevertheless, a three-satellite fix can only work if the GPS unit is synchronized with the atomic clocks onboard the satellites. Since it is impractical to install an atomic clock in every GPS unit (it would drive the cost of every unit to above $50,000), the designers of the GPS used an ingenious idea to correct for any timing errors between the (regular) clock in the GPS unit and the atomic clocks of the satellites: the distance to a fourth satellite is sought and a little trick of geometry (and algebra) is used to calculate the fix as well as to synchronize the GPS's internal clock with the atomic clocks in the satellites. The implication is that every GPS unit (yes, even your $100 hand-held unit) is also an atomic-accuracy clock. As mentioned earlier in the notes of chapter 4, one of the reasons for making the GPS system, which was developed and maintained by the U.S. military, accessible to commercial operators, was the navigation error that led to the downing of KAL 007. There are many similarities between the two accidents, which suggest that these navigational snafus are not only about the accuracy of the data, but also about user interaction with automated devices.

Chapter 9

Walk-in interfaces are pervasive and widespread. They are encountered on a daily basis by all of us wherever we turn, and not just by avid Internet surfers. We all use the telephone to call our bank, our doctor's office, the nearby box-office, or our insurance agent. We no longer hear a human voice on the other side of the line, but rather an automated answering system that guides (or misguides) us to our destination. This also is a "walk-in interface"—vocal rather than graphic. These automated voice answering systems often drive us up the wall. They tend to provide us with multiple and sometimes unclear options, offer menus that are annoyingly deep and multilayered, and when we finally understand where we missed the turn, give us no option for recovery. And when we just want to speak to a human on the other side of the line—not a chance. The system is set up to discourage us from reaching them. How often do we get frustrated and ultimately hang up just to retry our luck again?

There are four guiding principles that we can identify in a good walk-in interface: *Orientation*, *guidance*, *clarity of options* and, finally, *recoverability*. Orientation deals with where we are. A good interface must enable us to know, at all times, exactly where we are in the system (or interface). This is precisely the principle that was violated in the airliner entertainment system that we described earlier. The initial state remained obscure and the passenger had no idea what to do.

Guidance is about "where do we go from here?" or "how do we get there?" A good walk-in interface must enable us to know the answers to these questions from the interface or from our population stereotypes. It must not assume that we will learn by trial and error. If the designer wants to spare the trained and experienced users the added overhead, an override switch can be provided as an option for eliminating the guidance tips.

Clarity of options refers to description of choices. What can we, or must we, do next? The options must be clear and well defined. In walk-in interfaces, the user must not be assumed to have prior knowledge. He must be guided unambiguously by the interface. The internet visitor information system violated this second principle in that it did not specify clearly the implications of the various termination options.

Recoverability deals with what recourse is provided after making a mistake or taking the wrong turn. A good interface should minimize the damage. It must not leave us hanging there helplessly. It must not delete our previous work when we return from an unwanted detour and must guide us smoothly back to course. To conclude, while walk-in interfaces slowly mature and are getting progressively better, they are still frequently problematic for us. Designers are learning from gained experience. But not every mistake must be first made just to be corrected later.

Chapter 10

The strong, albeit reversal, relationship between the design of displays and magic is described in eloquent detail by Edward R. Tufte in his book *Visual Explanations: Images and Quantities, Evidence and Narrative* (Graphics Press, 1997). See chapter 3, "Explaining Magic: Pictorial Instructions and Disinformation Design" for a tour de force on this topic.

As another aspect of this theme, note that the first part of the chapter deals with how web designers can exploit population stereotypes to the advantage of the web-page owner rather than the user. The web page is designed to serve its owner, who is benefiting from "exploiting" the user. This role reversal is significant, since we have always assumed that an interface is designed to serve the user, and suddenly . . . woops—it's the owner who is being served! But indeed, this is what, for example, advertising is all about. We do not turn our TV on just to watch the "new and improved suds campaign," we came to watch our favorite police drama. But the advertiser foots the bill of the programs we like. Ultimately, the interface designer must serve the owner without losing the user—not a simple challenge.